Rendu basé image avec contraintes sur les gradients

Grégoire Nieto, Frédéric Devernay, James L. Crowley

Rendu basé image avec contraintes sur les gradients

15 juin 2016

Grégoire Nieto [1], **Frédéric Devernay** [2], **James Crowley** [3]

1. *Laboratoire Jean Kuntzmann, Univ. Grenoble Alpes, Inria Grenoble Rhône-Alpes, 655, avenue de l'Europe, F-38330 Montbonnot-Saint-Martin, France.*
   gregoire.nieto@inria.fr
2. *Laboratoire Jean Kuntzmann, Univ. Grenoble Alpes, Inria Grenoble Rhône-Alpes, 655, avenue de l'Europe, F-38330 Montbonnot-Saint-Martin, France.*
   frederic.devernay@inria.fr
3. *Laboratoire d'Informatique de Grenoble, Univ. Grenoble Alpes, Inria Grenoble Rhône-Alpes, 655, avenue de l'Europe, F-38330 Montbonnot-Saint-Martin, France.*
   james.crowley@inria.fr

RÉSUMÉ. *Le rendu basé image consiste à générer un nouveau point de vue à partir d'un ensemble de photos d'une scène. On commence en général par effectuer une reconstruction 3D approximative de la scène, utilisée par la suite pour synthétiser l'image cherchée à partir des images sources. Malheureusement, les discontinuités dans les poids des images sources, dues à la géométrie de la scène ou au placement des caméras, causent des artefacts visuels dans la vue résultante. Dans cet article nous montrons qu'une façon d'éviter ces artefacts est d'imposer des contraintes supplémentaires sur le gradient de l'image synthétisée. Nous proposons une approche variationnelle suivant laquelle l'image cherchée est solution d'un système linéaire résolu de façon itérative. Nous testons la méthode sur plusieurs jeux de données multi-vues structurés et non-structurés, et nous montrons que non seulement elle est plus performante que les méthodes de l'état de l'art, mais elle élimine aussi les artefacts créés par les discontinuités de visibilité.*

ABSTRACT. *Multi-view image-based rendering consists in generating a novel view of a scene from a set of source views. In general, this works by first doing a coarse 3D reconstruction of the scene, and then using this reconstruction to establish correspondences between source and target views, followed by blending the warped views to get the final image. Unfortunately,*

discontinuities in the blending weights, due to scene geometry or camera placement, result in artifacts in the target view. In this paper, we show how to avoid these artifacts by imposing additional constraints on the image gradients of the novel view. We propose a variational framework in which an energy functional is derived and optimized by iteratively solving a linear system. We demonstrate this method on several structured and unstructured multi-view datasets, and show that it numerically outperforms state-of-the-art methods, and eliminates artifacts that result from visibility discontinuities.

MOTS-CLÉS : *Rendu basé image, reconstruction 3D, imagerie computationelle.*

KEYWORDS: *Image-Based Rendering, 3D Reconstruction, Computational Photography.*

DOI:10.3166/HSP.x.1-26

1. Extended Abstract

Multi-view image-based rendering consists in generating a novel view of a scene from a set of source views. In general, this works by first doing a coarse 3D reconstruction of the scene, and then using this reconstruction to establish correspondences between source and target views. The final image can be obtained by warping and blending the input views : this is a direct rendering method. An alternative approach is to use a variational method, in which a smoothness term is used to regularize the solution. The expression of the energy is derived from the Bayesian formulation of the posterior : it is composed of a data term that forces the solution to fit the input images in intensity, and a smoothness term that accounts for what we know *a priori* about the solution, assumed to be a relatively smooth natural image. This energy functional is derived and optimized by iteratively solving a linear system.

The recent work of (Pujades *et al.*, 2014) derived new weights of contribution of each input image in the energy, that are based on the reconstructed scene geometry. Unfortunately, discontinuities in the these weights, due to imperfect geometry reconstruction or camera placement, result in artifacts in the target view. In this paper, we show how to avoid these artifacts by imposing additional constraints on the image gradients of the novel view. We propose a variational framework in which a new data term is added to the energy functional. This term forces the solution to be close to the data (the input views) in the gradient domain. It comes from the observation that a contour in the input images that is visible from the target view, should also appear in the image solution.

Our rendering method is generic, and could be applied to any camera placement. In particular we provide details on the derivation of the warps and the weights of contribution based on a set of depth maps that are generated from the input views thanks to an Multi-View Stereo algorithm. We demonstrate our variational rendering algorithm on several structured (*HCI Light Field Benchmark Datasets* and *Stanford Light Field Archive*) and unstructured (Strecha *et al.*, 2008) multi-view datasets, and show that it numerically outperforms state-of-the-art methods, and eliminates artifacts that result from visibility discontinuities. Moreover it enforces the *continuity* property that was first stated by (Buehler *et al.*, 2001) as a property that any ideal algorithm should satisfy.

2. Introduction

Le rendu basé image consiste à générer un nouveau point de vue à partir d'un ensemble de photos d'une scène. Il s'opère généralement en deux étapes successives: une pahse de reconstruction de la géométrie de la scène et une phase de synthèse de vue. On commence par effectuer un modèle 3D approximatif de la scène, appelé *proxy géométrique*. Un *proxy géométrique* est obtenu par un logiciel de MVS (*Multi-view Stereo*), et son format peut-être un nuage de points, un ensemble de cartes de profondeurs ou un maillage. Il permet d'estimer une fonction permettant d'aller des coordonnées en pixel dans l'image du point de vue cible aux coordonnées dans une image source, par projection du point de l'image cible sur le *proxy* puis reprojection dans l'image source (figure 1). Cette fonction est utilisée conjointement aux images sources pour synthétiser le point de vue voulu par une méthode directe ou variationnelle.

Le récent travail de (Pujades *et al*., 2014) propose une formulation bayésienne du problème de rendu basé image, construite sur le travail précédent de Wanner et Goldluecke (Goldluecke, Cremers, 2009 ; Wanner, Goldluecke, 2012b). C'est une méthode variationnelle, c'est-à-dire une méthode qui propose d'estimer l'image cherchée par minimisation d'une fonction de coût, une énergie. Chaque pixel de la solution est le résultat de la contribution, directe ou indirecte, de plusieurs pixels des vues sources. Ces contributions sont pondérées par des termes qui apparaissent dans l'expression de l'énergie. Ils ont montré que les poids des pixels des images sources dans l'énergie pouvaient se déduire formellement des propriétés de la caméra, du contenu de l'image, et de la précision du *proxy géométrique*, amenant une nouvelle formalisation des heuristiques de poids proposées initialement par (Buehler *et al*., 2001). La plupart des « propriétés désirables qu'un algorithme idéal de rendu basé image devrait avoir » (Buehler *et al*., 2001) eurent alors une explication formelle, sauf la propriété de *continuité*. En effets les contributions des vues sources varient brutalement d'un pixel de la vue cible à l'autre, soit parce que la limite du champ de vue d'une caméra est atteinte ou qu'elle est occultée (sa contribution tombe à 0), soit parce que les contributions ne sont pas lisses au sein même de la même image source, puisque l'estimation d'un *proxy géométrique* est bruitée et que le calcul des poids des contributions repose la reconstruction du *proxy*.

Dans cet article, nous montrons qu'une façon d'éviter ces artefacts est d'imposer des contraintes supplémentaires sur le gradient de l'image synthétisée. Ces contraintes viennent d'une simple observation: les contours d'image dans la vue cible doivent aussi être des contours dans les images sources où ces parties sont visibles. Une fonctionnelle d'énergie similaire à celle de (Pujades *et al*., 2014) est développée, composée de l'habituel terme sur les données (*data term*) et un terme de régularisation (*smoothness term*), mais le terme sur les données contient un terme additionnel qui prend en compte les contraintes sur les gradients. Nous montrons que tenir compte à la fois de l'intensité et du gradient dans les méthodes de rendu basé image apporte une solution élégante au renforcement de la propriété de *continuité* initialement énoncée par (Buehler *et al*., 2001).

3. Travaux antérieurs

3.1. Rendu basé image (IBR)

Les techniques de rendu basé image ont été décrites et classifiées par (Shum *et al.*, 2008). La plupart des méthodes de l'état de l'art (Kopf *et al.*, 2013 ; Sinha *et al.*, 2012 ; Lipski *et al.*, 2014 ; 2010 ; Chaurasia *et al.*, 2013) utilisent une reconstruction de la géométrie de la scène plus ou moins précise, appelée géométrie intermédiaire ou *proxy géométrique*. (Ortiz-Cayon *et al.*, 2015) proposent de segmenter les images en super-pixels et de calculer la qualité de plusieurs algorithmes d'IBR afin de choisir le meilleur pour chaque super-pixel. Toutes ces techniques sont inspirées par la méthode directe de (Buehler *et al.*, 2001), qui effectuent une combinaison des k plus proches vues, pondérées par les angles et les distances à la vue cible, garantissant ainsi un champ de mélange lisse. La continuité du mélange résultant dans le domaine image est certifiée par le renforcement du lissage spatial de ces poids, mais des artefacts temporels sont toujours présents si l'ensemble des caméras qui contribuent est trop éparse. (Davis *et al.*, 2012) proposent une technique de rendu par subdivisions des points de vue sources pour créer un meilleur champ de mélange. Néanmoins, les poids des contributions sont encore calculés d'après des règles heuristiques et le choix des caméras pour le rendu est totalement arbitraire.

3.2. But d'une approche variationnelle

Le but d'une approche variationnelle comme celle de (Wanner, Goldluecke, 2012b) est d'estimer une image u – une fonction qui à tout pixel, ou point image 2D, renvoie une couleur – à partir des données, les k images sources appelées v_k. L'estimateur $\hat{u}$ de la solution u doit maximiser la probabilité *a posteriori* d'observer l'image cherchée u sachant nos données en entrée v_k et la probabilité *a priori* de u : on l'appelle estimateur MAP (*Maximum a posteriori*). Ne sachant pas calculer cette probabilité *a posteriori* nous l'exprimons autrement à l'aide du théorème de Bayes en fonction de la vraisemblance et des probabilités *a priori*. En prenant nos probabilités indépendantes et normalement distribuées, on arrive à la conclusion que la solution $\hat{u}$ doit minimiser la somme de deux termes E_{color} et E_{prior}. La méthode est aussi qualifiée d'inverse, car ne sachant pas exprimer la solution en fonction des données, on va plutôt chercher à exprimer les données en fonction de la solution, supposée connue.

3.3. Se débarrasser des heuristiques

Se débarrasser des heuristiques et du réglage manuel des paramètres est une idée clé de (Wanner, Goldluecke, 2012b). La contribution de chaque vue dans l'estimation de la solution est automatiquement déduite d'équations mathématiques. (Pujades *et al.*, 2014) vont plus loin en intégrant l'incertitude géométrique dans le formalisme bayésien. Ils obtiennent alors de nouveaux poids qui favorisent les caméras satisfaisant à la fois la cohérence épipolaire (*epipole consistency*) et la déviation angulaire mini-

male (*minimal angular deviation*), deux principes établis par (Buehler *et al.*, 2001) pour décrire l'algorithme d'IBR idéal. Cependant leur méthode n'offre pas de cadre formel pour satisfaire le principe de continuité (*continuity principle*), en particulier près des limites du champ de vue des caméras. Nous montrons qu'introduire un terme additionnel dans la fonctionnelle énergie, qui contraint non seulement les intensités mais aussi les gradients de la solution, apporte une solution élégante au principe de continuité.

3.4. Rendu en haute résolution

L'idée clé pour un rendu en haute résolution est que la qualité de l'image solution dépend souvent de la contrainte de l'espace de recherche. Par conséquent, trouver la bonne régularisation ou *a priori* sur la solution est une question cruciale pour l'obtention d'images d'excellente qualité. La contribution principale de (Fitzgibbon *et al.*, 2003) est l'utilisation d'*a priori* calculés à partir de grandes bases de données de textures afin de contraindre la solution, indépendamment des données relatives à la scène. Cette idée fut récemment étendue par (Flynn *et al.*, 2015) qui effectuent une synthèse de nouvelle vue à partir d'un réseau de neurones, entraîné par une gigantesque base d'images prises tout autour du monde. Au contraire notre méthode ne repose pas sur de forts *a priori* sur la nouvelle image à synthétiser, mais tente plutôt de mieux exploiter les données procurées par les images sources pour ajouter de nouvelles contraintes sur la solution. En somme notre algorithme ne requiert pas ces énormes bases de données pour produire des images de haute qualité. Cependant les deux approches ne sont pas incompatibles, et on pourrait imaginer intégrer notre méthode à celle de (Flynn *et al.*, 2015).

3.5. Fusion d'images dans le domaine du gradient

La fusion d'images dans le domaine du gradient a reçu beaucoup d'intérêt ces dernières années, en commençant par l'article phare de (Pérez *et al.*, 2003), pour des applications dans l'édition d'images (McCann, Pollard, 2008), l'*inpainting* (Levin *et al.*, 2003) ou les panoramas (Agarwala *et al.*, 2004 ; Zomet *et al.*, 2006). Le travail le plus proche du notre en rendu basé image est probablement celui de (Kopf *et al.*, 2013), qui effectue un rendu de gradient, suivi d'une intégration pour produire la couleur de l'image. Néanmoins cette méthode se limite l'interpolation de points de vue, et elle ne traite pas des configurations génériques, dans lesquelles les points de vue sources sont très différents et non structurés. Notre méthode s'attelle à ce problème en proposant un cadre plus générique pour le rendu basé image multi-vues.

4. Aperçu général de l'approche

L'objectif de notre méthode est de synthétiser une nouvelle image optimale u au point de vue cible à partir des images sources v_k. Les images sont des fonctions définies sur des sous-ensembles continus de $\mathbb{R}^2$ qui renvoient une intensité ou une cou-

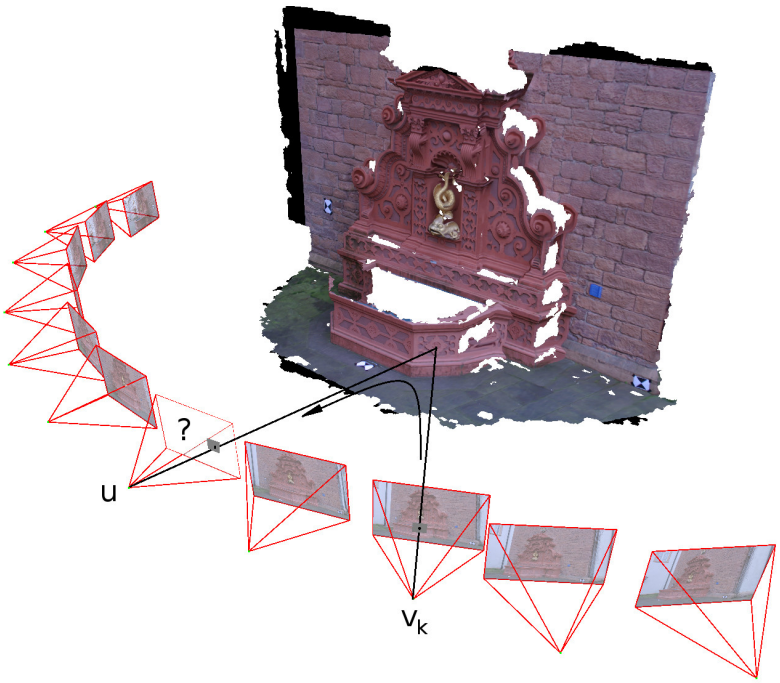

Figure 1. La reconstruction 3D de la scène permet de mettre en correspondance la vue cible u avec les vues sources v_k.

leur : $u : \Gamma \subset \mathbb{R}^2 \to \mathbb{R}$ et $v_k : \Omega_k \subset \mathbb{R}^2 \to \mathbb{R}$. Dans un souci de simplicité, les valeurs des images sont supposées scalaires, mais il serait aisé de généraliser aux images couleurs $v_k : \Omega_k \to \mathbb{R}^3$. Dans toutes nos expériences nous traitons les images dans l'espace couleur RGB. La discrétisation de ces fonctions est abordée dans la section 5.1.

Notre méthode de rendu-basé image est décomposable en deux étapes indépendantes : la reconstruction 3D (section 6) et le rendu (section 5).

Reconstruction 3D Le but de cette étape est recaler les images sources via les *warps* τ_k qui transforment tout point $\mathbf{x}_m = (x_m, y_m)$ de la vue source k en son correspondant $\mathbf{x}_p = (x_p, y_p)$ dans la vue cible :

$$\begin{array}{rccc} \tau_k : & \Omega_k & \to & \Gamma \\ & \mathbf{x}_m & \mapsto & \mathbf{x}_p \end{array} \tag{1}$$

Dans un premier temps les images sources sont utilisées afin de calibrer les caméras pour extraire leurs matrices de paramètres intrinsèques et extrinsèques. Puis un algorithme de stéréo multi-vues produit un *proxy géométrique*, en l'occurrence une carte de profondeur par vue source, afin d'estimer les déformations d'images (*warps*) pour les projeter sur la vue cible. Définies telles quelles, les fonctions de déformation

ne sont pas bijectives : des points image dans les vues sources n'ont pas de projeté dans la vue cible parce que la projection sort du champ de vue cible et des points de la vue cible n'ont pas de projeté inverse dans une vue source à cause des auto-occultations. La gestion de ces deux types d'occultations se fait par l'estimation d'une carte de profondeur du point de vue ciblé, et permet de retirer les points occultés des domaines Ω_k et Γ. À l'issue de cette étape, les *warps* définis sur les nouveaux domaines sont bijectifs. Enfin les paramètres des caméras sources, combinés aux différentes cartes de profondeur, nous permettent de calculer les poids des contributions de chaque vue dans l'étape de rendu.

Rendu Une fois obtenus les poids de contribution et les correspondances par pixel avec la vue ciblée, on peut procéder à l'étape de rendu à proprement parler. Nous développons une formulation variationnelle du modèle de rendu basé image de l'état de l'art, que nous étendons par l'ajout de contraintes sur les gradients des images sources.

Dans la section 7, nous testons la méthode sur plusieurs jeux de données multi-vues structurés et non-structurés, et nous montrons que non seulement elle est plus performante que les méthodes de l'état de l'art, mais elle élimine aussi les artefacts créés par les discontinuités de visibilité.

5. Une formulation variationnelle du rendu basé image

5.1. Modèle de formation de l'image

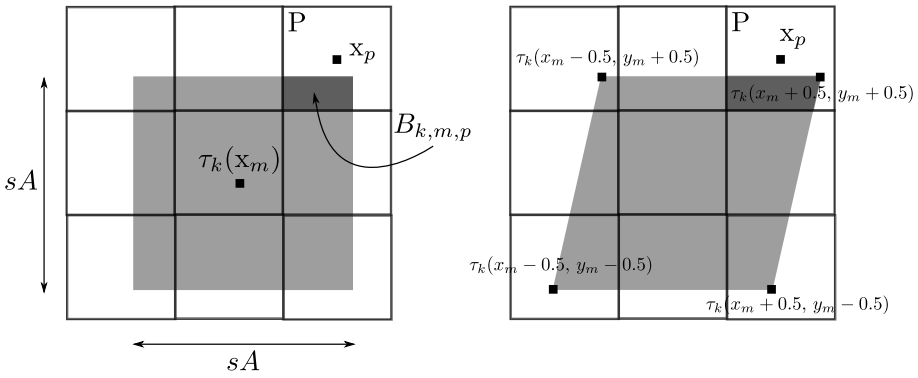

Figure 2. L'aire de la projection du pixel est colorée en gris. À gauche : Nous supposons que la transformée de la PSF est toujours carrée ; l'intensité résultante est la moyenne des valeurs de l'image cible pondérée par les coefficients $B_{k,m,p}$ – l'intersection (en gris foncé) entre l'aire en gris et le pixel en haute résolution. À droite : Modèle plus précis où nous supposons seulement que le warp *est localement linéaire et que la PSF transformée est obtenue en transformant chaque coin du pixel source.*

Comme on suppose en général dans la littérature sur la super-résolution (Baker, Kanade, 2002 ; Hardie *et al*., 1997), la valeur d'intensité $v_k(\mathbf{x}_m)$ d'un point $\mathbf{x}_m$ dans l'image source k peut s'écrire comme la convolution de l'image cible avec la fonction d'étalement du point ou PSF [1], notée b. Étant donnée une image idéale u au point de vue cible, définie sur le domaine Γ, et une fonction de déformation τ_k des points de Ω_k dans Γ, si nous mettons de côté les occultations pour le moment, l'intensité de l'image observée peut s'écrire comme la relation de convolution

$$v_k(\mathbf{x}_m) = \int_{\Omega_k} u \circ \tau_k(\mathbf{x})\, b(\mathbf{x} - \mathbf{x}_m)\, \mathrm{d}\mathbf{x}, \tag{2}$$

ou plus simplement $v_k = b * (u \circ \tau_k)$.

La PSF $b : \Omega_k \to [0, 1]$ est la densité de probabilité qui peut s'écrire $b_k : \Gamma \to [0, 1]$ par changement de variable $\mathbf{x}' = \tau_k(\mathbf{x})$ de telle façon que

$$v_k(\mathbf{x}_m) = \int_{\Gamma} u(\mathbf{x}') b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\, \mathrm{d}\mathbf{x}'. \tag{3}$$

Il y a plusieurs manières de calculer la PSF transformée b_k, selon comment on modèle la PSF initiale b. L'hypothèse la plus commune est de considérer la PSF comme étant une gaussienne 2D d'espérance $\mathbf{x}_m$ et de covariance $\mathbf{\Sigma}$. Un modèle plus simple de la PSF est de se représenter un pixel carré et uniformément sensible à la lumière, la PSF étant alors une densité uniforme. En notant A l'aire du pixel centré sur $(0, 0)$ dans une vue source k, on obtient

$$b(x, y) = \begin{cases} \frac{1}{A^2} & \text{if } -\frac{1}{A} \leq x, y \leq \frac{1}{A} \\ 0 & \text{sinon.} \end{cases} \tag{4}$$

Sous l'hypothèse que le *warp* τ_k est localement linéaire, la PSF transformée est un parallélogramme uniformément distribué (figure 2). Dans ce cas, on peut faire une hypothèse encore plus forte et supposer que le *warp* préserve les pixels (leur aire et leur forme), ce qui est faux en réalité mais simplifie grandement l'implémentation. Désormais nous prendrons des pixels d'aire unitaire. Puisque l'intensité est constante et égale à $u(\mathbf{p})$ sur toute la surface du pixel $\mathbf{p}$ dans la vue cible, la relation de convolution (2) ci-dessus peut être écrite comme l'ont fait (Hardie *et al*., 1997) :

$$v_k(\mathbf{x}_m) = \sum_{\mathbf{p} \in \Gamma} u(\mathbf{p}) \int_{\mathbf{p}} b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\, \mathrm{d}\mathbf{x}', \tag{5}$$

et l'intensité du pixel dans l'image source est

$$v_k(\mathbf{m}) = \sum_{\mathbf{p} \in \Gamma} B_{k,m,p} u(\mathbf{p}), \tag{6}$$

1. *Point Spread Function*

où $B_{k,m,p} = \int_{\mathbf{p}} b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\,\mathrm{d}\mathbf{x}'$ est l'aire de l'intersection entre la projection du pixel dans la vue cible et le pixel $\mathbf{p}$ de cette même vue. Si l'échantillonnage des vues de départ et d'arrivée sont les mêmes, alors les aires d'intersection sont les coefficients bilinéaires : c'est équivalent à interpoler bilinéairement les intensités de u.

5.2. Estimation du maximum a posteriori

Le but de l' approche variationnelle est d'estimer une image u à partir des données $(v_k^*)_{k\in[1..K]}$, où K est le nombre de vues sources. L'estimateur $\hat{u}$ de la solution u doit maximiser la probabilité *a posteriori* ; on l'appelle estimateur MAP (Maximum *a posteriori*) :

$$\hat{u} = \arg\max_u P(u|(v_k^*)_{k\in[1..K]}). \tag{7}$$

Ne sachant pas calculer cette probabilité *a posteriori* nous l'exprimons autrement à l'aide du théorème de Bayes en fonction de la vraisemblance et des probabilités *a prori*. On fait l'hypothèse que les v_k sont conditionnellement indépendants. Le terme $P((v_k^*)_{k\in[1..K]})$, appelé *évidence*, ne dépend pas de u et peut donc être retiré de l'équation :

$$\hat{u} = \arg\max_u \frac{P((v_k^*)_{k\in[1..K]}|u)\,P(u)}{P((v_k^*)_{k\in[1..K]})} = \arg\max_u \prod_{k\in[1..K]} P((v_k^*)|u)\,P(u). \tag{8}$$

Le terme de vraisemblance $P((v_k^*)_{k\in[1..K]}|u)$ est la probabilité d'obtenir les données (images sources) supposant connue la solution u. Elle s'exprime comme le produit des probabilités $P(v_k^*|u)$ dont la loi est prise normale : $P(v_k^*|u) \propto e^{-E_{\mathrm{color},k}(u)}$ $E_{\mathrm{color},k(u)}$ est un terme aux moindres carrés qui représente la somme des écarts aux images sources en terme d'intensité (couleur du pixel). Il est aussi appelé terme d'attache aux données dans la littérature :

$$E_{\mathrm{color},k}(u) = \frac{1}{2}\int_{\Omega_k} \omega_k(u)\,(b * (u\circ\tau_k) - v_k^*)^2\,\mathrm{d}\mathbf{x}. \tag{9}$$

Le termes $\omega_k(u)$ sont les contributions par pixel de chaque image source. Ils dépendent du gradient de la solution courante u et de l'incertitude sur la géométrie reconstruite. Une formule explicite de ces contributions (34) est donnée dans la section 6. En supposant que le bruit du capteur est gaussien et identique pour toutes les images, nous notons $\sigma_{s,k}^2 = \lambda$ sa variance, une constante strictement positive. La vraisemblance étant le produit des $P(v_k^*|u)$, elle s'exprime

$$P((v_k^*)_{k\in[1..K]}|u) \propto e^{-\frac{1}{\lambda}E_{\mathrm{color}}(u)},\ \text{avec } E_{color}(u) = \sum_{k=1}^{K} E_{\mathrm{color},k}(u). \tag{10}$$

La probabilité *a priori* $P(u)$ représente notre connaissance *a priori* sur l'image à synthétiser. Nous savons que cette dernière est naturelle, donc comporte peu de

variations de gradient : le signal est *régulier*. Dans notre travail, nous utilisons un *a priori* de variation totale (Goldluecke, Cremers, 2010), norme L^1. Une preuve de convergence est fournie par (Chambolle, 2004).

$$P(u) \propto e^{-E_{\text{prior}}(u)}, \text{ avec } E_{prior}(u) = \int_\Gamma |\nabla u|. \tag{11}$$

On rappelle que l'estimateur MAP $\hat{u}$ doit minimiser la probabilité *a posteriori* :

$$\begin{aligned}
\hat{u} &= \arg\max_u P(u|(v_k^*)_{k\in[1..K]}) && (12)\\
&= \arg\min_u -\ln P(u|(v_k^*)_{k\in[1..K]}) && (13)\\
&= \arg\min_u -\ln P((v_k^*)_{k\in[1..K]}|u) - \ln P(u) && (14)\\
&= \arg\min_u \frac{1}{\lambda} E_{color}(u) + E_{prior}(u). && (15)
\end{aligned}$$

Le paramètre λ strictement positif permet de contrôler la prépondérance du terme de régularisation dans l'énergie. Notre solution $\hat{u}$ doit minimiser

$$E(u) = E_{color}(u) + \lambda E_{prior}(u). \tag{16}$$

5.3. Ajout du terme portant sur le gradient de l'image

Figure 3. Les discontinuités des warps τ_k *et des poids* ω_k *provoquent des artefacts. À gauche: la vue globale d'une scène de Strecha estimée avec l'énergie donnée par l'équation 16. En haut à droite: un zoom révèle des artefacts haute-fréquence. En bas à droite : une carte de profondeur présentant des discontinuités dues à la visibilité à cet endroit de l'image.*

Les cartes de profondeur estimées sont bruitées et incomplètes. Le bruit fait référence à la variance élevée et aux nombreux *outliers* des carte de profondeur, dû à

l'incertitude de localisation. Il peut être corrigé par un lissage ou un seuillage, au pris d'une perte de précision et d'une plus grande sparsité du modèle. Les discontinuités sont dues aux limites des champs de vue des caméras et aux auto-occultations. Par conséquent les $\omega_k(u)$ dans (9) et les τ_k peuvent être aussi bruités et discontinus, ce qui résulte en des artefacts dans la solution finale qui apparaissent comme de faux bords ou textures (figure 3). La méthode de rendu basé image devrait empêcher ces contours d'apparaître : en fait, un contour synthétisé dans l'image solution devrait également être présent dans les images sources, là où ces parties de la scène sont visibles.

Pour renforcer cette propriété, nous ajoutons un terme supplémentaire $E_{grad}(u)$, qui force la solution courante à se rapprocher des données dans le domaine du gradient. Ce terme permet en outre d'ajouter de nouvelles contraintes au système qui est alors mieux conditionné. Nous l'obtenons à partir de $P((\nabla v_k^*)_{k\in[1..K]}|\nabla u)$, la probabilité d'obtenir les gradients des images sources, sachant le gradient de l'image cible. Nous supposons que les variables aléatoires ∇v_k sont indépendantes, identiquement distribuées et obéissent à une loi normale, d'où

$$\begin{aligned} E_{grad}(u) &\propto -\ln P((\nabla v_k^*)_{k\in[1..K]}|\nabla u) && (17)\\ &\propto -\sum_{k=1}^{K} \ln P(\nabla v_k^*|\nabla u) && (18)\\ &= \sum_{k=1}^{K} \frac{1}{2} \int_{\Omega_k} (\nabla v_k - \nabla v_k^*)^2 \,\mathrm{d}\mathbf{x} && (19)\\ &= \sum_{k=1}^{K} \frac{1}{2} \int_{\Omega_k} (\nabla (b * (u \circ \tau_k)) - \nabla v_k^*)^2 \,\mathrm{d}\mathbf{x}. && (20) \end{aligned}$$

Trouver u qui minimise ce terme d'énergie particulier est alors équivalent à résoudre un système de K équations de Laplace :

$$\Delta(b * (u \circ \tau_k) - v_k^*) = 0, \tag{21}$$

où $\Delta = \nabla.\nabla$ représente le laplacien de l'image. On en déduit alors la différentielle de la fonctionnelle :

$$\mathrm{d}E_{grad}(u) = (|\frac{\partial \tau_k}{\partial z}|^{-1}\, \bar{b} * (\Delta(b * (u \circ \tau_k)) - \Delta v_k^*)) \circ \beta_k. \tag{22}$$

Les β_k sont les déformations inverses qui apparaissent à cause du changement de variable dans l'intégrale. $\bar{b}$ est l'adjoint de la PSF b. Les déformations τ_k sont celles qui ont été estimées auparavant, et manquent donc de précision. Cette incertitude a un effet néfaste sur le calcul de $\Delta(b * (u \circ \tau_k))$. Par conséquent, nous choisissons de calculer le laplacien de u d'abord, puis de le transformer dans le domaine Ω_k. Sous l'hypothèse que les déformations τ_k sont localement linéaires, on peut négliger leurs dérivées au deuxième ordre et obtenir

$$\Delta(b * (u \circ \tau_k)) = b * \left(\frac{\partial \tau_k}{\partial x}^{\intercal} \mathbf{H}u \frac{\partial \tau_k}{\partial x} + \frac{\partial \tau_k}{\partial y}^{\intercal} \mathbf{H}u \frac{\partial \tau_k}{\partial y} \right), \tag{23}$$

où $\mathbf{H}u = \frac{\partial \nabla u}{\partial \mathbf{x}}$ est la hessienne de u.

Les cartes de profondeur mal estimées causant de fortes discontinuités dans les correspondances entre les vues, la hessienne peut être très instable. Pour l'implémentation et dans ce cas seulement, nous supposons que $\tau_k(\mathbf{x}) \approx \mathbf{x} + d$, de telle façon que

$$\Delta(b * (u \circ \tau_k)) = b * (\mathrm{trace}(\mathbf{H}u) \circ \tau_k) = b * (\Delta u \circ \tau_k). \tag{24}$$

La forme finale de l'énergie à minimiser est donc

$$E(u) = \alpha E_{color}(u) + \gamma E_{grad}(u) + \lambda E_{prior}(u). \tag{25}$$

5.4. Discrétisation

A ce stade il est légitime de se demander comment passer d'une formulation continue du problème à une solution numérique. Pour chaque pixel $\mathbf{m}$ de chaque vue source k nous obtenons une équation similaire à (6). Soit $\mathbf{V}^*$ le vecteur de tous les pixels de toutes les vues sources mis dans une seule grande colonne $(v_0(0), v_0(1), ..., v_{K-1}(M-1))$, $\mathbf{U}$ le vecteur colonne contenant la solution courante $(u(0), ..., u(N-1))$, et $\mathbf{B}$ la matrice $KM \times N$ qui contient les coefficients $B_{k,m,p}$. On peut donc écrire naturellement $\mathbf{V} = \mathbf{BU}$. Par conséquent on exprime l'énergie (10) comme un système linéaire:

$$E_{color}(\mathbf{U}) = (\mathbf{BU} - \mathbf{V}^*)^\intercal \mathbf{W} (\mathbf{BU} - \mathbf{V}^*), \tag{26}$$

où $\mathbf{W}$ est une matrice $KM \times KM$ diagonale qui contient les poids $|J_{\mathbf{x}'}(\beta_k)|\omega_k$. Pour minimiser cette énergie nous dérivons le système linéaire, et obtenons les équations normales qui nous apportent un estimateur de la solution $\hat{\mathbf{U}}$:

$$\mathbf{B}^\intercal \mathbf{W} \mathbf{B} \hat{\mathbf{U}} = \mathbf{B}^\intercal \mathbf{W} \mathbf{V}^*. \tag{27}$$

La matrice $\mathbf{B}^\intercal \mathbf{W} \mathbf{B}$ n'est en général pas inversible. Le système linaire peut être résolu par n'importe quelle méthode des moindres carrés linéaires.

De même que le terme sur les couleurs de l'image, le terme portant sur les gradients est

$$E_{grad}(\mathbf{U}) = (\mathbf{B}\nabla\mathbf{U} - \nabla\mathbf{V}^*)^\intercal (\mathbf{B}\nabla\mathbf{U} - \nabla\mathbf{V}^*) \tag{28}$$

et se dérive identiquement. Nous minimisons la fonctionnelle (25) ainsi discrétisée via FISTA (*Fast Iterative Shrinkage Thresholding Algorithm*) (Beck, Teboulle, 2009).

6. Reconstruction 3D

La plupart des méthodes de rendu basé image utilisent une reconstruction 3D approximative de la scène appelée *proxy géométrique*. Nous avons opté pour une représentation en cartes de profondeur car elles sont un bon compromis entre précision et exhaustivité de reconstruction. En effet, la reconstruction d'un nuage de points à l'aide d'un algorithme de l'état de l'art (Furukawa, Ponce, 2010) est économe en données

et très précise mais les données sont éparses. D'autre part, si une reconstruction de surface (Kazhdan *et al*., 2006 ; Fuhrmann, Goesele, 2014) est faite à partir du nuage de points dans le but de densifier les correspondances entre les vues, la précision de la géométrie diminue. Les cartes de profondeur offrent en outre l'avantage d'établir immédiatement les correspondances τ_k entre tout point $\mathbf{x}_m$ d'une vue source v_k son projeté $\mathbf{x}_p$ sur la vue cible u.

6.1. Coordonnées homogènes

On notera $\bar{\mathbf{x}} = (x, y, 1)$ le point image $\mathbf{x} = (x, y)$ auquel on a rajouté une troisième coordonnée, en unités pixel. De la même façon on note $\tilde{\mathbf{x}} = z.\bar{\mathbf{x}} = z.(x, y, 1)$ le point homogène associé. Le passage des coordonnées homogènes en coordonnées euclidiennes se fait par le biais de la fonction de normalisation N_e telle que $N_e(\tilde{\mathbf{x}}) = \mathbf{x}$. Elle divise un point en coordonnées homogènes par sa dernière composante, ici la profondeur orthogonale z du point 3D correspondant. Le passage de coordonnées euclidiennes en coordonnées étendues 3D se fait via la fonction N_h telle que $N_h(\mathbf{x}) = \bar{\mathbf{x}}$ par ajout d'une troisième composante. La jacobienne de N_h est constante, en revanche celle de N_e dépend des coordonnées du point homogène. Ces jacobiennes de normalisation en coordonnées homogènes sont données par (Heuel, 2004) à la page 110. Nous invitons le lecteur à consulter cet ouvrage pour de plus amples informations sur la géométrie projective dans le contexte de la propagation d'incertitude.

6.2. Calibration des caméras

Nous utilisons une partie du logiciel de reconstruction 3D multi-vues MVE (Fuhrmann *et al*., 2014). Dans un premier temps, nous corrigeons la distorsion radiale des caméras et les calibrons à l'aide d'openMVG (Moulon *et al*., 2013). Le modèle sténopé est alors choisi pour représenter les caméras: une matrice 3×3 de paramètres intrinsèques $\mathbf{K}_k$, ainsi que la matrice 3×3 de rotation $\mathbf{R}_k$ et le vecteur de translation 3D $\mathbf{t}_k$ permettant le changement en coordonnées monde/caméra. Le centre optique de chaque caméra peut être calculé à partir des paramètres extrinsèques: $\mathbf{C}_k = -\mathbf{R}_k^\intercal \mathbf{t}_k$. S'il l'on appelle $\mathbf{X}_m$ le point 3D associé au point homogène $\tilde{\mathbf{x}}_m \in \Omega_k$ situé à une distance orthogonale z_m de la caméra k. On a alors $\tilde{\mathbf{x}}_\mathbf{m} = \mathbf{K}_k(\mathbf{R}_k\mathbf{X}_m + \mathbf{t}_k)$ ou encore en coordonnées monde $\mathbf{X}_m = \mathbf{R}_k^\intercal \mathbf{K}_k^{-1}\tilde{\mathbf{x}}_\mathbf{m} + \mathbf{C}_k$.

Nous supposons connus les paramètres de la vue à synthétiser $\mathbf{K}_u$, $\mathbf{R}_u$, $\mathbf{t}_u$ et $\mathbf{C}_u$.

6.3. Calcul des cartes de profondeur

Pour chaque vue k, une carte de profondeur est estimée en utilisant l'algorithme de stéréo multi-vues (Goesele *et al*., 2007). Les profondeurs obtenues h sont radiales

– distances euclidiennes entre un point 3D de la scène et le centre de la caméra. Nous les convertissons en profondeurs orthogonales:

$$z_m = \frac{h_m}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m\|}. \tag{29}$$

En effet en coordonnées locales à la caméra k, le point 3D s'écrit $\mathbf{X}_m = \mathbf{K}_k^{-1}\tilde{\mathbf{x}}_m = z_m.\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m$. Ainsi en supposant que les distances sont en valeur absolue, on obtient $h_m = \|\mathbf{X}_m\| = z_m.\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m\|$ d'où l'égalité (29).

De la même façon, la dérivée spatiale $h_{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}}$ donnant l'orientation de la surface peut être convertie en

$$z_{\mathbf{x},m} = \frac{\partial z_m}{\partial \mathbf{x}} = \frac{1}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m\|}(h_{\mathbf{x},m} - h_m.\frac{(\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m)^\intercal(\mathbf{K}_k^{-1}[0]\,\mathbf{K}_k^{-1}[1])}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m\|^2}) \tag{30}$$

où $\mathbf{K}_k^{-1}[0]$ et $\mathbf{K}_k^{-1}[1]$ représentent respectivement la première et la deuxième colonne de $\mathbf{K}_k^{-1}$.

Chaque carte de profondeur est filtrée par un filtre bilatéral (Kopf *et al.*, 2007) dans le but de combler les trous.

6.4. Correspondances par pixel

Les déformations d'images τ_k sont calculées en projetant sur la vue cible le point 3D estimé par la carte de profondeur de la vue source k. Le schéma suivant permet de se rendre compte des différentes opérations qui opèrent lors de la transformation d'un point image d'une caméra à l'autre:

$$\tau_k : \quad \mathbf{x_m} \quad \underset{(a)}{\longrightarrow} \quad \bar{\mathbf{x}}_m \quad \underset{(b)}{\longrightarrow} \quad \tilde{\mathbf{x}}_\mathbf{m} \quad \underset{(c)}{\longrightarrow} \quad \mathbf{X_m} \quad \underset{(d)}{\longrightarrow} \quad \tilde{\mathbf{x}}_\mathbf{p} \quad \underset{(e)}{\longrightarrow} \quad \mathbf{x_p} \tag{31}$$

(a) Le point image 2D $\mathbf{x_m}$ (en unités pixel) est étendu en $\bar{\mathbf{x}}_m$ par ajout d'une troisième coordonnée valant 1 via la fonction N_h. (b) La multiplication par la distance orthogonale z_m vue de la caméra k nous donne le point homogène $\tilde{\mathbf{x}}_\mathbf{m}$. (c) Un changement de repère par le biais des matrices de la caméra k permet d'obtenir les coordonnées monde du point 3D $\mathbf{X_m} = \mathbf{R_k^\intercal K_k^{-1}}\tilde{\mathbf{x}}_\mathbf{m} + \mathbf{C_k}$. (d) Puis le point homogène $\tilde{\mathbf{x}}_\mathbf{p}$ est obtenu par changement inverse dans le repère de la caméra ciblée $\tilde{\mathbf{x}}_\mathbf{p} = \mathbf{K}_u(\mathbf{R}_u\mathbf{X}_m + \mathbf{t}_u)$. (e) Enfin le point image $\mathbf{x_p}$ résultant (en unités pixels) est calculé par normalisation euclidienne $N_e(\tilde{\mathbf{x}}_\mathbf{p})$. Nous avons donc

$$\tau_k(\mathbf{x}_m) = N_e(\mathbf{K}_u(\mathbf{R}_u(z_m.\mathbf{R}_k^\intercal\mathbf{K}_k^{-1}N_h(\mathbf{x}_m) + \mathbf{C}_k) + \mathbf{t}_u)). \tag{32}$$

6.5. Gestion des occultations

Le traitement des cartes de profondeur permet d'estimer les fonctions de déformation τ_k. Telles quelles elles ne sont pas bijectives : un point dans une vue source

Figure 4. La carte de profondeur de la vue cible, obtenue par projection de quads depuis les vues sources. Images de la base fountain*.*

k, projeté dans la vue cible par τ_k n'est peut-être pas visible dans cette vue (auto-occultation). Il nous faut donc réduire le domaine de départ (Ω_k) aux points qui sont visibles dans la vue cible ; c'est le rôle de l'étape de gestion des occultations. À l'issue de cette dernière, on a retiré des domaines de départ les pixels où les *warps* ne sont pas définis à cause des occultations, les τ_k sont bijectifs sur les domaines résultants.

La gestion de la visibilité se fait en deux temps. D'abord, les pixels des vues sources sont marqués invalides si leur projeté par τ_k se situe hors des bords de l'image de la vue cible. Ensuite, la carte de profondeur z_u de la vue cible est estimée pour traiter les occultations inverses (figure 4). À partir de chaque pixel $\mathbf{x}_m$ de chaque vue source v_k un *quad* (quadrilatère 3D) est créé à la distance estimée z_m et orienté par $z_{\mathbf{x}}(\mathbf{x}_m)$. Il est ensuite projeté sur la vue cible u en accumulant un *z-buffer* pour ne retenir que les profondeurs les plus proches. Le test de visibilité s'effectue en comparant la distance du point 3D reconstruit à la vue cible avec la profondeur estimée précédemment: si la différence se situe au-delà d'un certain seuil – fixé arbitrairement – alors le pixel $\mathbf{x}_m$ est marqué comme non visible depuis u.

6.6. Contributions des vues sources

Les poids des contributions de chaque vue source dans le rendu sont de deux sortes (Pujades *et al*., 2014): les poids de résolution et de déformation.

Les poids de déformation, donnés par le jacobien $|\frac{\partial \tau_k}{\partial \mathbf{x}_m}|$ de la déformation τ_k, pénalisent les vues qui observent la surface de biais ou qui se situent loin de la vue

cible. La jacobienne se calcule à partir des matrices des caméras, des profondeurs et des normales estimées, en dérivant en chaîne la fonction composée (32) par rapport au point image 2D $\mathbf{x}_m$. Dérivons alors chacune des déformations intermédiaires énoncées dans la chaîne (32). (e) $\frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_m} = J_e(\tilde{\mathbf{x}}_p)\frac{\partial \tilde{\mathbf{x}}_p}{\partial \mathbf{x}_m}$. (d) $\frac{\partial \tilde{\mathbf{x}}_p}{\partial \mathbf{x}_m} = \mathbf{K}_u\mathbf{R}_u\frac{\partial \mathbf{X}_m}{\partial \mathbf{x}_m}$. (c) $\frac{\partial \mathbf{X}_m}{\partial \mathbf{x}_m} = \mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\frac{\partial \tilde{\mathbf{x}}_m}{\partial \mathbf{x}_m}$. (b) $\frac{\partial \tilde{\mathbf{x}}_m}{\partial \mathbf{x}_m} = z_m.\frac{\partial \bar{\mathbf{x}}_m}{\partial \mathbf{x}_m} + \bar{\mathbf{x}}_m z_{\mathbf{x},m}$. (a) $\frac{\partial \bar{\mathbf{x}}_m}{\partial \mathbf{x}_m} = J_h$. Ces équations misent bout-à-bout, on obtient

$$\frac{\partial \tau_k}{\partial \mathbf{x}_m} = J_e(\tilde{\mathbf{x}}_p)\mathbf{K}_u\mathbf{R}_u\mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\begin{pmatrix} z_{x,m}x_m + z_m & z_{y,m}x_m \\ z_{x,m}y_m & z_{y,m}y_m + z_m \\ z_{x,m} & z_{y,m}\end{pmatrix}. \quad (33)$$

Les poids de géométrie, garantissant la *déviation angulaire minimale*, c'est-à-dire pénalisant les caméras formant un angle trop grand avec la vue cible, découlent de l'incertitude de la géométrie estimée $\sigma_{g,k}^2$ et de la variance du bruit du capteur $\sigma_{s,k}^2$. Chaque vue source k est ainsi pondérée par $\omega_k(u) = (\sigma_{s,k}^2 + \sigma_{g,k}^2(u))^{-1}$ avec $\sigma_{g,k}^2(u) = \sigma_{z,k}^2(b\star(\frac{\partial \tau_k}{\partial z}^\intercal\nabla u\circ\tau_k))^2$, et ∇u le gradient de la solution courante u. De la même façon que pour les poids de résolution, la dérivés $\frac{\partial \tau_k}{\partial z}$ s'obtient par composition des dérivées de la chaîne (32). (e) $\frac{\partial \mathbf{x}_p}{\partial z_m} = J_e(\tilde{\mathbf{x}}_p)\frac{\partial \tilde{\mathbf{x}}_p}{\partial z_m}$. (d) $\frac{\partial \tilde{\mathbf{x}}_p}{\partial z_m} = \mathbf{K}_u\mathbf{R}_u\frac{\partial \mathbf{X}_m}{\partial \mathbf{x}_m}$. (c) $\frac{\partial \mathbf{X}_m}{\partial \mathbf{x}_m} = \mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\frac{\partial \tilde{\mathbf{x}}_m}{\partial z_m}$. (b) $\frac{\partial \tilde{\mathbf{x}}_m}{\partial z_m} = z_m.\frac{\partial \bar{\mathbf{x}}_m}{\partial z_m} + \bar{\mathbf{x}}_m$. (a) $\frac{\partial \bar{\mathbf{x}}_m}{\partial z_m} = \mathbf{0}_2$. Enfin mis bout-à-bout

$$\frac{\partial \tau_k}{\partial z_m} = J_e(\tilde{\mathbf{x}}_p)\mathbf{K}_u\mathbf{R}_u\mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m. \quad (34)$$

7. Expériences

Afin de mettre en évidence l'influence du terme sur les gradients des images sur la qualité de la vue synthétisée, plusieurs expériences ont été conduites sur des scènes synthétiques et réelles, pour des placements de caméras structurés ou non.

Il est important de préciser que les parties de la vue cible qui ne sont visibles par aucune des vues sources sont remplies par un algorithme de va-et-vient (*push/pull* inpainting de (Gortler *et al.*, 1996)). L'effet provoqué est celui d'un remplissage par diffusion de ces zones (figures 5 et 6).

7.1. Base de données structurées

Les premières expériences ont été réalisées à partir d'une base de données d'images *light-field* (Wanner *et al.*, 2013), prises du *HCI Light Field Benchmark Datasets* et de la *Stanford Light Field Archive*. Pour chaque base d'images nous utilisons une matrice de vues adjacentes (3×3 ou 1×9). Nous appliquons la méthode de (Wanner, Goldluecke, 2012a) pour estimer les disparités entre les vues et l'incertitude de la géométrie à partir des huit vues voisines. La vue centrale est rendue par chaque algorithme testé puis comparée avec l'image originale qui sert de référence pour évaluer les performances de l'algorithme. Toutes les images sources sont prises en compte dans la synthèse de point de vue.

On rappelle que α, β et γ sont des réels positifs qui pondèrent chaque terme de l'énergie (17), respectivement le terme d'attache aux données sur les couleurs, le terme d'attache aux données sur les gradients et le terme de régularisation. α et λ sont fixés à leur valeur d'origine dans les expériences précédentes (Wanner, Goldluecke, 2012b ; Pujades *et al*., 2014), respectivement 1.0 et 0.1. Nous faisons varier γ de 0 à 3 pour observer l'influence du terme sur les gradients. Un γ nul est bien entendu équivalent à minimiser la même fonctionnelle que celle de (Pujades *et al*., 2014), mais notre implémentation diffère légèrement de la leur, ce qui explique pourquoi nous avons représenté les deux dans le tableau 7.1. Nous comparons également notre méthode à celle de (Wanner, Goldluecke, 2012b). Toutes les expériences sont réalisées sur carte graphique (nVidia GTX Titan). La résolution du système prend entre 2 et 3 secondes pour des images de résolution 768×768.

Le PSNR (*Peak Signal to Noise Ratio*, plus il est haut mieux c'est) et le $\mathrm{DSSIM} = 10^4(1 - \mathrm{SSIM})$ (Wang *et al*., 2004) (*Structural dissimilarity*, plus il est faible mieux c'est) sont calculés par rapport à la vue de référence pour évaluer nos résultats. Le deuxième jeu d'expériences a été réalisé avec les mêmes images, mais avec une géométrie plane – la disparité estimée est constante, ce qui correspond à un plan, le *proxy géométrique* le plus grossier. L'incertitude de la géométrie est augmentée.

Notre terme sur les gradients améliore systématiquement les résultats avec une disparité estimée, et très souvent pour une disparité plane. La qualité des images générées est une fonction croissante de γ. Nous interprétons ceci par le fait que le terme sur les gradients ajoute de nouvelles contraintes au système, permettant à l'algorithme d'optimisation une meilleure convergence vers le minimum global de l'énergie.

7.2. Base de données non structurées

Les expériences suivantes (figure 7) ont été réalisées sur des vues réelles prises de la base de données de (Strecha *et al*., 2008), *fountain* et *herzjesu*, ainsi que sur nos propres jeux de données *charce* et *lion*. Les résultats numériques du tableau 1 viennent confirmer les figures d'illustration en terme d'efficacité de l'ajout du terme portant sur les gradients des images.

Comme le système est faiblement contraint, des hautes fréquences apparaissent dans les zones visibles depuis peu de caméras. Ces artefacts sont accentués par une estimation très bruitée de la profondeur près des régions d'occultation (autour du poisson de la fontaine, ou du bas-relief de Jésus). Le paramètre λ contrôlant le terme de régularisation *Total Variation* a été augmenté à 0.003 pour réduire l'apparition de ces hautes fréquences. Mais le résultat est peu convainquant car les images perdent alors du détail comparées aux originales. Nous avons alors baissé λ pour conserver tous les traits du bas-relief et ajouté le terme sur les gradients ($\gamma = 1.0$). Nous pouvons voir sur les images, quelle que soit la géométrie utilisée pour le rendu, que les artefacts disparaissent tout en préservant les détails de l'image. Le terme sur les couleurs est conservé pour que la couleur originale des images ne soit pas affectée mais mis à une valeur faible ($\alpha = 0.1$). L'ajout du terme sur les gradients a en outre permis d'empê-

Résultats numériques sur les bases d'images synthétiques et réelles (Wanner *et al.*, 2013). Notre méthode est comparée à celle de (Wanner, Goldluecke, 2012b) et de (Pujades *et al.*, 2014). Le *proxy géométrique* est soit estimé par (Wanner, Goldluecke, 2012b) soit mis à profondeur constante avec une grande incertitude. Pour chaque *light-field*, la première valeur est le PSNR (plus il est aussi mieux c'est), la seconde est 10^{-4} fois le DSSIM. $\text{DSSIM} = 10^4(1 - \text{SSIM})$ (Wang *et al.*, 2004) (plus il est faible mieux c'est). La meilleure performance est en gras. Voir la section 6.2 pour de plus amples détails sur l'expérience.

	HCI light fields, raytraced				HCI light fields, gantry				Standford light fields, gantry					
	buddha		stillLife		maria		couple		truck		gum nuts		tarot	
Disparité estimée														
SAVSR(Wanner, Goldluecke, 2012b)	42.84	17	30.13	58	40.06	53	26.55	226	33.75	408	31.82	1439	28.71	60
BVS(Pujades *et al.*, 2014)	42.37	18	30.45	55	40.10	53	28.50	178	33.78	**407**	31.93	1437	28.88	58
$\gamma = 0.0$	43.07	15	30.75	50	39.91	53	32.93	92	33.73	434	31.98	1430	25.37	51
$\gamma = 1.0$	43.28	14	30.83	49	40.14	51	33.05	90	33.82	430	32.08	1428	25.58	48
$\gamma = 2.0$	43.43	13	30.86	49	40.36	48	33.15	88	33.91	427	32.17	1426	25.74	46
$\gamma = 3.0$	**43.59**	**12**	**30.90**	**48**	**40.55**	**46**	**33.22**	**87**	**33.98**	423	**32.25**	**1424**	**25.89**	**44**
Disparité plane														
SAVSR(Wanner, Goldluecke, 2012b)	34.28	74	21.28	430	31.65	144	20.07	725	32.48	419	30.55	1403	22.64	278
BVS(Pujades *et al.*, 2014)	37.51	44	22.24	380	34.38	99	**22.88**	**457**	33.79	386	31.30	1378	23.78	218
$\gamma = 0.0$	37.69	42	**22.27**	**377**	34.28	100	22.74	468	34.50	367	31.38	1359	24.47	189
$\gamma = 1.0$	37.74	41	**22.27**	378	34.34	97	22.74	468	34.57	365	31.39	1359	24.51	187
$\gamma = 2.0$	37.80	**40**	22.26	378	34.40	95	22.74	468	34.66	362	**31.43**	1358	24.50	187
$\gamma = 3.0$	**37.84**	**40**	22.25	378	**34.45**	**93**	22.74	468	**34.68**	**360**	31.42	**1357**	**24.58**	**185**

Figure 5. Résultats sur les jeux de données fountain *et* herzjesu*. La colonne de droite montre un échantillon des images sources.*

cher l'apparition de faux contours près des frontières de visibilité, garantissant ainsi la propriété de *continuité*.

Les résultats numériques mettent en évidence que l'amélioration de la qualité des images synthétisées sauf dans le cas du jeu de données *charce*. En effet les meilleurs résultats sont obtenus avec un fort terme de lissage, alors que l'ajout du terme sur les gradients n'influe quasiment pas. Précisons que la reconstruction 3D de cette scène est très mauvaise du fait de la diversité des points de vues utilisés et de la présence du ciel, arrière-plan non texturé. Par conséquent de nombreuses zones ne sont pas reconstruites, et l'ensemble des pixels sur lesquels nous pouvons évaluer la qualité des résultats (l'ensemble des pixels reconstruits) est trop restreint pour que l'évaluation soit pertinente. En outre on peut penser que dans le cas où la reconstruction est extrêmement bruitée, une forte stabilisation par le terme de lissage est plus bénéfique que l'ajout du terme d'attache aux gradients.

Figure 6. Résultats sur les jeux de données charce *et* lion. *La colonne de droite montre un échantillon des images sources.*

8. Discussion et conclusion

Nous avons présenté une méthode de rendu basé image qui permet de générer une nouvelle vue à partir d'un ensemble générique et non structuré d'images. Cette méthode est inspirée par les travaux de (Pujades *et al.*, 2014), qui ont oeuvré pour formaliser la plupart des « propriétés désirables » listées dans l'article phare de (Buehler *et al.*, 2001). Leur approche fut d'introduire une formulation bayésienne du problème de rendu et d'obtenir la vue cherchée par un processus d'optimisation. La seule propriété qu'ils n'ont pu formaliser fut la propriété de *continuité*, qui énonce que les contributions de chaque vue source doivent être des fonctions continues des coordonnées des pixels.

Nous avons montré qu'un moyen de garantir cette *continuité* est de déclarer que les contours, textures et détails ne devraient pas être créés dans l'image cible s'ils ne

image de référence	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.002$	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.003$	$\alpha = 0.1, \gamma = 1.0,$ $\lambda = 0.002$

Figure 7. Rendu avec différents jeux de paramètres $(\alpha, \gamma, \lambda)$ qui contrôlent la proportion des différents termes dans la formule de l'énergie (25). Les deux premières lignes montrent des résultats sur la collection d'images fountain*, et les deux dernières sur la collection* herzjesu*. L'approche proposée est $\gamma \neq 0$.*

	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.002$ (Pujades *et al*., 2014) (Wanner, Goldluecke, 2012b)		$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.003$ (Pujades *et al*., 2014) (Wanner, Goldluecke, 2012b)		$\alpha = 0.1, \gamma = 1.0,$ $\lambda = 0.002$ (notre méthode)	
fountain – view 2	21.03	132	21.09	120	**21.16**	**107**
fountain – view 5	26.00	74	26.14	64	**26.36**	**51**
fountain – view 8	22.00	140	22.08	125	**22.16**	**111**
herzjesu – view 2	21.73	186	**21.96**	153	21.93	**143**
herzjesu – view 4	23.13	194	23.81	130	**23.90**	**115**
herzjesu – view 6	18.08	349	18.26	287	**18.31**	**273**
charce – view 4	13.74	905	**13.85**	**885**	13.73	901
charce – view 6	9.953	1360	**10.04**	**1317**	10.00	1352
charce – view 11	14.20	859	**14.38**	**793**	14.20	843
lion – view 1	24.40	190	24.36	198	**24.46**	**175**
lion – view 3	29.16	103	29.15	107	**29.57**	**87**
lion – view 5	22.18	298	22.25	294	**22.27**	**277**

TABLE *1. Résultats numériques sur des bases de données non structurées. Notre méthode est comparée aux méthodes de l'état de l'art (Pujades* et al*., 2014 ; Wanner, Goldluecke, 2012b), pour lesquelles il n'y a pas de contraintes sur les gradients de l'image ($\gamma = 0.0$). Pour chaque résultat, la première valeur est le PSNR (plus il est élevé mieux c'est), la seconde étant le DSSIM (plus il est faible mieux c'est).* $\mathrm{DSSIM} = 10^4(1 - \mathrm{SSIM})$. *La meilleure valeur est notée en gras.*

sont pas présents dans les images sources aux endroits visibles. Cela implique l'ajout d'un terme additionnel portant sur les données sources, basé sur les gradients des images. L'énergie ainsi modifiée peut être minimisée en résolvant de façon itérative un système linéaire dérivé de la fonctionnelle. Ce système est alors plus contraint et mieux conditionné que le précédent, ce qui empêche l'apparition d'artefacts près des frontières de visibilité. Ce résultat montre une nette amélioration par rapport aux précédentes méthodes de rendu basées sur les intensités, à la fois en terme de mesure qualitative et en terme de qualité subjective.

Cette méthode pourrait être retravaillée pour optimiser directement les gradients de la vue cible, plutôt que les intensités ; puis l'intensité de l'image pourrait être reconstruite en résolvant l'équation de Poisson, comme il est fait par (Kopf *et al*., 2013). Cela devrait complètement enlever toutes les variations dans l'image synthétisée qui viennent de discontinuités des fonctions de visibilité, qui sont toujours visibles dans nos résultats, bien qu'atténuées.

Remerciements

Nous remercions la DGA pour nous avoir co-financé sur ce travail de recherche.

Bibliographie

Agarwala A., Dontcheva M., Agrawala M., Drucker S., Colburn A., Curless B. *et al*. (2004). Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*, p. 294–302. New York, NY, USA, ACM.

Baker S., Kanade T. (2002). Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n° 9, p. 1167–1183.

Beck A., Teboulle M. (2009). A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, p. 693–696.

Buehler C., Bosse M., McMillan L., Gortler S., Cohen M. (2001). Unstructured lumigraph rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, p. 425–432. New York, NY, USA, ACM.

Chambolle A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, vol. 20, n° 1-2, p. 89–97.

Chaurasia G., Duchene S., Sorkine-Hornung O., Drettakis G. (2013). Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.*, vol. 32, n° 3, p. 30:1–30:12.

Davis A., Levoy M., Durand F. (2012). Unstructured light fields. *Computer Graphics Forum*, vol. 31, n° 2pt1, p. 305–314.

Fitzgibbon A., Wexler Y., Zisserman A. (2003). Image-based rendering using image-based priors. In *IEEE ICCV, 2003. proceedings*, p. 1176–1183 vol.2.

Flynn J., Neulander I., Philbin J., Snavely N. (2015). DeepStereo: Learning to predict new views from the world's imagery.

Fuhrmann S., Goesele M. (2014). Floating scale surface reconstruction. *ACM Transactions on Graphics (TOG)*, vol. 33, n° 4, p. 46.

Fuhrmann S., Langguth F., Goesele M. (2014). MVE – a multiview reconstruction environment. In *Proceedings of the eurographics workshop on graphics and cultural heritage (gch)*, vol. 6, p. 8.

Furukawa Y., Ponce J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, n° 8, p. 1362–1376.

Goesele M., Snavely N., Curless B., Hoppe H., Seitz S. M. (2007). Multi-view stereo for community photo collections. In *Computer vision, 2007. iccv 2007. ieee 11th international conference on*, p. 1–8.

Goldluecke B., Cremers D. (2009). Superresolution texture maps for multiview reconstruction. In *2009 IEEE 12th International Conference on Computer Vision*, p. 1677–1684.

Goldluecke B., Cremers D. (2010). An approach to vectorial total variation based on geometric measure theory. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 327–333.

Gortler S. J., Grzeszczuk R., Szeliski R., Cohen M. F. (1996). The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, p. 43–54. New York, NY, USA, ACM.

Hardie R., Barnard K., Armstrong E. (1997). Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, vol. 6, n° 12, p. 1621–1633.

Heuel S. (2004). *Uncertain projective geometry: Statistical reasoning for polyhedral object reconstruction*. Springer Science & Business Media.

Kazhdan M., Bolitho M., Hoppe H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, p. 61–70. Aire-la-Ville, Switzerland, Switzerland, Eurographics Association.

Kopf J., Cohen M. F., Lischinski D., Uyttendaele M. (2007). Joint bilateral upsampling. In *ACM SIGGRAPH 2007 Papers*. New York, NY, USA, ACM.

Kopf J., Langguth F., Scharstein D., Szeliski R., Goesele M. (2013). Image-based rendering in the gradient domain. *ACM Trans. Graph.*, vol. 32, n° 6, p. 199:1–199:9.

Levin A., Zomet A., Weiss Y. (2003). Learning how to inpaint from global image statistics. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, p. 305–312 vol.1.

Lipski C., Klose F., Magnor M. (2014). Correspondence and depth-image based rendering a hybrid approach for free-viewpoint video. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, n° 6, p. 942–951.

Lipski C., Linz C., Berger K., Sellent A., Magnor M. (2010). Virtual video camera: Image-based viewpoint navigation through space and time. *Computer Graphics Forum*, vol. 29, n° 8, p. 2555–2568.

McCann J., Pollard N. S. (2008). Real-time gradient-domain painting. In *ACM SIGGRAPH 2008 Papers*, p. 93:1–93:7. New York, NY, USA, ACM.

Moulon P., Monasse P., Marlet R. (2013). La bibliothèque openMVG: open source multiple view geometry. In *Orasis, congrès des jeunes chercheurs en vision par ordinateur*.

Ortiz-Cayon R., Djelouah A., Drettakis G. (2015). A bayesian approach for selective image-based rendering using superpixels. In *3d Vision (3dv), International Conference on*. IEEE.

Pérez P., Gangnet M., Blake A. (2003). Poisson image editing. In *Acm siggraph 2003 papers*, p. 313–318. New York, NY, USA, ACM.

Pujades S., Devernay F., Goldluecke B. (2014). Bayesian view synthesis and image-based rendering principles. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 3906–3913.

Shum H.-Y., Chan S.-C., Kang S. B. (2008). *Image-based rendering*. Springer Science & Business Media.

Sinha S. N., Kopf J., Goesele M., Scharstein D., Szeliski R. (2012). Image-based rendering for scenes with reflections. *ACM Trans. Graph.*, vol. 31, n° 4, p. 100.

Strecha C., Hansen W. von, Van Gool L., Fua P., Thoennessen U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008*, p. 1–8.

Wang Z., Bovik A., Sheikh H., Simoncelli E. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, vol. 13, n° 4, p. 600-612.

Wanner S., Goldluecke B. (2012a). Globally consistent depth labeling of 4D light fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 41–48.

Wanner S., Goldluecke B. (2012b). Spatial and angular variational super-resolution of 4D light fields. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), *Computer Vision – ECCV 2012*, p. 608–621. Springer Berlin Heidelberg.

Wanner S., Meister S., Goldluecke B. (2013). Datasets and benchmarks for densely sampled 4D light fields. In *Vision, Modelling and Visualization (VMV)*.

Zomet A., Levin A., Peleg S., Weiss Y. (2006). Seamless image stitching by minimizing false edges. *IEEE Transactions on Image Processing*, vol. 15, n° 4, p. 969–977.